THAT WHICH IS CLAIMED IS:

1. Method of handling instructions within a processor with decoupled architecture, the processor including a core containing several processing units (DU, AU, GU) associated respectively with memory means of the FIFO type for sequentially storing the respective instructions which are intended for the corresponding units, in which method the issuing within the core processor of an instruction for loading memory stored data into a register causes the issuing to the memory means of a first processing unit (DU) of an instruction for loading into said register and the issuing, to the memory means of a second processing unit (AU) of an instruction intended to derive said memory stored data, the loading instruction being executed only when said memory stored data has been issued by said second processing unit, characterized in that the memory means of the first processing unit includes a first FIFO-type memory (RLDQ), and a second FIFO-type memory (DIDQ) separate from the first one, in that each loading instruction (LDRx) is stored in the first memory (RLDQ) and at least some of the other operative instructions (INST) intended for the first unit (DU) are stored in the second memory, in that an operative instruction involving at least one register (DUOPx) and having reached the head (ETG4) of the second memory (DIDQ) is extracted from this second memory (DIDQ), if no loading instruction (LDRx) which is earlier in time and intended to alter the value of the register or registers associated with this operative instruction is present in the first memory, and in that, in the presence of such an earlier, load-modifying instruction (LDRx), said operative instruction (DUOPx) is extracted from the second memory (DIDQ) only after the load-modifying instruction (LDRx) has been extracted from the first memory (RLDQ).

2. Method according to claim 1, characterized in that, every time an instruction is stored in the second memory (DIDQ) without provision being made simultaneously to store a loading instruction (LDRx) in the first memory (RLDQ), a non-operative instruction (NOP) is stored in the first memory (RLDQ), in that, every time an instruction (LDRx, NOP) is extracted from the first memory (RLDQ) a first read counter (RDCTR1) is incremented, in that, every time an instruction (INST) is stored in the first memory (RLDQ), a first write counter (WRCTR1) is incremented, in that, every time a load instruction (LDRx) is stored in the first memory (RLDQ), the current value of the first write counter (WRVL1) is stored in memory, and in that the determining of the still-present character of this loading instruction (LDRx) in the first memory takes account of the result of the comparison of said memory stored current value of the first write counter with the current value of the first read counter (RDVL1).

3. Method according to claim 2, characterized in that the first write counter (WRCTR1) and the first read counter (RDCTR1) have an identical binary size which is equal to the depth of the first memory (RLDQ), in that an overflow bit (WROVFL1, RDOVFL1) is associated with each first counter, changing value every time the corresponding first counter comes back to its initial value, in that, every time a load instruction (LDRx) is stored in the first memory (RLDQ), the current value of the overflow bit (WROVFL1) of the first write counter is likewise stored in memory, and in that the determination of the still-present character of this load instruction (LDRx) in the first memory also takes into account the result of the comparison of the current value of the overflow bit (RDOVFL) of the first read counter with said memory

stored value of the overflow bit of the first write counter.

4. Method according to claim 2 or 3, characterized in that, every time an operative instruction (DUOPx) involving at least one register (Rx) is stored in the second memory (DIDQ), a first label is associated with it containing the current value of the first write counter which was stored in memory when the last load instruction (LDRx) modifying at least one of the registers involved with said operative instruction was stored in the first memory, and in that the criterion for extraction of this operative instruction (DUOPx) having reached the head (ETG4) of the second memory (DIDQ) then takes into account the result of the comparison between said memory stored current value associated with this operative instruction and the current value of the first read counter (RDVL1).

5. Method according to claims 3 and 4, characterized in that, every time an operative instruction is stored in the second memory, the current value of the overflow bit of the first write counter which was stored in memory when the last load instruction (LDRx) modifying at least one of the registers involved with said operative instruction was stored in the first memory is associated with it, also in its first label, and in that the criterion for extraction of this operative instruction having reached the head of the second memory also takes into account the result of the comparison between this memory stored current value associated with this operative instruction and the current value of the overflow bit (RDOVFL1) of the first read counter.

6. Method according to one of the preceding

claims, characterized in that the memory means of the first unit includes a third FIFO-type memory (SNDGQ), separate from the first and from the second memories, and a fourth FIFO-type memory (RCGQ), separate from the first three, in that the first unit (DU) contains a guard-indication register (RG), in that the issuing of a guarded instruction to the memory means (AIDQ) of another processing unit different from the first one (AU or GU), or the issuing of a guarded loading instruction to the first unit, causes the issuing to the first processing unit (DU), and storage in the third memory (SNDGQ), of a transmission instruction (SNDGx) intended to cause transmission to said other unit (AU or GU) or to said fourth memory (RCGQ), respectively, of the value of the guard indication (Gx) associated with said guarded instruction or with said guarded loading instruction, respectively, in that a transmission instruction (SNDGx) having reached the head (ETG4) of this third memory is then extracted from the third memory (SNDGQ), if no modifying instruction (GMIx), earlier in time and intended to modify the value of the guard indication (Gx) associated with this transmission instruction (SNDGx), is present in the second memory (DIDQ), and in that, in the presence of such an earlier modifying instruction (GMIx), the transmission instruction (SNDGx) is extracted from the third memory (SNDGQ) only after the modifying instruction (GMIx) has been extracted from the second memory (DIDQ).

7. Method according to claim 6, characterized in that, every time an instruction (INST) is extracted from the second memory (DIDQ), a second read counter (RDCTR2) is incremented, in that, every time an instruction (INST) is stored in the second memory (DIDQ), a second write counter (WRCTR2) is incremented, in that, every time an instruction (GMIx)

modifying the value of a guard indication (Gx) is stored in the second memory (DIDQ), the current value (WRVL2) of the second write counter is stored in memory, and in that the determining of the still-present character of a modifying instruction (GMIx) in the second memory takes account of the result of the comparison of said memory stored current value of the second write counter with the current value of the second read counter (RDVL2).

8. Method according to claim 7, characterized in that the second write counter (WRCTR2) and the second read counter (RDCTR2) have an identical binary size equal to the depth of the second memory (DIDQ), in that an overflow bit (WROVFL2, RDOVFL2) changing value every time the second corresponding counter comes back to its initial value is associated with each second counter, and in that, every time an instruction (GMIx) modifying the value of a guard indication is stored in the second memory, the current value of the overflow bit (WROVFL2) of the second write counter is likewise stored in memory, and in that the determining of the still-present character of this modifying instruction (GMIx) in the second memory also takes into account the result of the comparison of the current value of the overflow bit (RDOVFL2) of the second read counter with said memory stored value of the overflow bit of the second write counter.

9. Method according to claim 7 or 8, characterized in that, every time a transmission instruction (SNDGx) is stored in the third memory (SNDGQ), a label is associated with it containing the current value of the second write counter which was stored in memory when the last instruction (GMIx) modifying the guard indication (Gx) associated with said transmission instruction was stored in the second

memory, and in that the criterion for extraction of this transmission instruction (SNDGx) having reached the head (ETG4) of the third memory (SNDGQ) takes into account the result of the comparison between said memory stored current value associated with this transmission instruction and the current value of the second read counter (RDVL2).

10. Method according to claims 8 and 9, characterized in that, every time a transmission instruction is stored in the third memory, the current value of the overflow bit of the second write counter which was stored in memory when the last instruction (GMIx) modifying the guard indication associated with said transmission instruction was stored in the second memory is associated with it, also in its label, and in that the criterion for extraction of this transmission instruction having reached the head of the third memory likewise takes into account the result of the comparison between this memory stored current value associated with this transmission instruction and the current value of the overflow bit (RDOVFL2) of the second read counter.

11. Processor with decoupled architecture, comprising a core containing several processing units (DU, AU, GU), associated respectively with FIFO-type memory means for sequentially storing the respective instructions which are intended for the corresponding units, and a central unit (CU) able, in response to an instruction for loading memory stored data into a register, to issue to the memory means of a first processing unit (DU) an instruction for loading into said register, and to issue to the memory means of a second processing unit (AU) an instruction intended to derive said memory stored data, the loading instruction being executed only when said memory stored data has

been delivered by said second processing unit, characterized in that the memory means of the first unit includes a first FIFO-type memory (RLDQ) and a second FIFO-type memory (DIDQ) separate from the first one, and in that the processor includes:

routing means (DISP), able to store each loading instruction (LDRx) in the first memory (RLDQ) and to store at least some of the other operative instructions (INST) intended for the first unit (DU) in the second memory, and

first control means (MCTL1) capable of extracting from the second memory (DIDQ) an operative instruction (DUOPx) involving at least one register (DUOPx) and having reached the head (ETG4) of this second memory, if no earlier loading instruction LDRx intended to modify the value of the register or registers associated with this operative instruction is present in the first memory (DIDQ), and

in the presence of such an earlier load-modifying instruction (LDRx), of extracting said operative instruction (DUOPx) from the second memory (SNDGQ) only after the load-modifying instruction (LDRx) has been extracted from the first memory (RLDQ).

12. Processor according to claim 11, characterized in that the first control means (MCTL1) include:

a first read counter (RDCTR1) incremented every time an instruction is extracted from the first memory (RLDQ),

a first write counter (WRCTR1) incremented every time an instruction is stored in the first memory (RLDQ),

a set of first individual registers (RT0-RT15) associated respectively with the set of registers (R0-R15) capable of being loaded by memory stored data,

a first control unit (MC00) able, whenever an

instruction (LDRx) for loading into a register (Rx) is stored in the first memory (RLDQ), to store the current value (WRVL1) of the first write counter in a main field (LCHx) of the first individual register (Rx) associated with this register (Rx),

a second control unit (MC11) able to determine the still-present character of this instruction (LDRx) for loading into the first memory (RLDQ), and including first comparison means (MC100) able to compare the content of the main field (LCHx) of the first individual register (RTx) with the current value (RDVL1) of the first read counter.

13. Processor according to claim 12, characterized in that the first write counter and the first read counter have an identical binary size equal to the depth of the first memory (RLDQ), in that an overflow bit (WROVFL1, RDOVFL1), changing value every time the first corresponding counter comes back to its initial value, is associated with each first counter, in that this first individual register (RTx) further includes a one-bit auxiliary field (LBAx), in that the first control unit (MC00) is able, every time an instruction (LDRx) for loading into a register (Rx) is stored in the first memory (RLDQ), also to store the current value of the overflow bit (WROVFL1) of the first write counter in the auxiliary field (LBAx) of the first corresponding individual register (RTx), and in that the second control unit (MC11) includes first auxiliary comparison means (PLO10) able to compare the current value of the overflow bit (RDOVFL1) of the first read counter with the content of the auxiliary field (LBAx).

14. Processor according to claim 13, characterized in that the first auxiliary comparison means include an EXCLUSIVE NOR logic gate (PLO10).

15. Processor according to one of claims 12 to 14, characterized in that each stage (LETGi) of the second memory (DIDQ) includes a field (LCHUi) which is usable for the storage of an operative instruction (DUOPx) involving at least one register (Rx) or else for the storage of a non-operative instruction (NOP), and a second supplementary field (LDtag), in that the control means include:

a third control unit (MC12) able, every time an operative instruction (DUOPx) involving at least one register (Rx) is stored in the usable field (LCHU1) of the input stage of the second memory, to select, from among all the first individual registers associated with all the registers involved in said operative instruction, the one corresponding to the last loading instruction (LDRx) modifying at least one of these registers involved, and to transfer the content of the main field (LCHx) of the first selected individual register (RTx), into the first supplementary field (LDtag) of the input stage of the second memory (DIDQ), and

a fourth control unit (MC13) able to derive the criterion for extraction of this operative instruction (DUOPx) having reached the head stage (LETG4) of the second memory (DIDQ), and including first comparison means (MC300) able to compare the content of said first supplementary field (LDtag) of the header stage with the current value (RDVL1) of the first read counter.

16. Processor according to claim 15 taken in combination with claim 13 or 14, characterized in that each stage of the second memory further includes a second supplementary field (AuxLD), in that the third control unit (MC12) is able, every time an operative instruction is stored in the usable field (LCHU1) of the input stage of the second memory, to transfer the

content of the auxiliary field (LBAx) of the first individual register (RTx) selected, into the second supplementary field (AuxLD) of the input stage of the second memory, and in that the fourth control unit (MC13) includes first supplementary comparison means (PLO30) able to compare the content of the second supplementary field (AuxLD) with the current value of the overflow bit (RDOVFL1) of the first read counter.

17. Processor according to claim 16, characterized in that the first supplementary comparison means include an EXCLUSIVE NOR logic gate (PLO30).

18. Processor according to one of claims 11 to 17, characterized in that the memory means of the first unit (DU) includes a third FIFO-type memory (SNDGQ), separate from the first and from the second memories, and a fourth FIFO-type memory (RCGQ), separate from the first three, in that the first unit (DU) contains a guard-indication register (GR), in that the central unit (CU) is able

to issue a guarded instruction to the memory means (AIDQ) of another processing unit (GU) different from the first one, or a guarded loading instruction to the first processing unit (DU), and

to issue to the memory means of the first processing unit (DU) a transmission instruction (SNDGx) intended to have the value of the guard indication (Gx) associated with said guarded instruction or with said guarded loading instruction respectively transmitted to said other unit (GU) or to said fourth memory (RCGQ), respectively,

in that the router means (DISP) are able to store each transmission instruction (SNDGx) in the third memory (SNDGQ),

and in that the processor further includes

second control means able

to extract from the third memory (SNDGQ) a transmission instruction (SNDGx) having reached the head (ETG4) of this third memory if no modifying instruction (GMIx) which is earlier in time and intended to modify the value of the guard indication associated with this transmission instruction is present in the second memory (DIDQ), and

in the presence of such an earlier modifying instruction (GMIx), to extract the transmission instruction from the third memory (SNDGQ) only after the modifying instruction (GMIx) has been extracted from the second memory (DIDQ).

19. Processor according to claim 18, characterized in that the second control means (MCTL) include

a second read counter (RDCTR2) incremented every time an instruction is extracted from the second memory,

a second write counter (WRCTR2) incremented every time an instruction is stored in the second memory,

a set of second individual registers (GT0-GT15) associated respectively with the set of guard indications (G0-G15),

a fifth control unit (MC0) able, every time an instruction (GMIx) modifying the value of a guard indication (Gx) is stored in the second memory, to store the current value (WRVL2) of the second write counter in a main field (CHx) of the second individual register (GTx) associated with this guard indication,

a sixth control unit (MC1) able to determine the still-present character of this modifying instruction (GMIx) in the second memory, and including second comparison means (MC10) able to compare the content of the main field (CHx) of the second

individual register (GTx) with the current value (RDVL2) of the second read counter.

20. Processor according to claim 19, characterized in that the second write counter and the second read counter have an identical binary size equal to the depth of the second memory (DIDQ), in that an overflow bit (WROVFL2, RDOVFL2) changing value every time the second corresponding counter comes back to its initial value is associated with each second counter, in that each second individual register (GTx) further includes an auxiliary, one-bit field (BAx), in that the fifth control unit (MC0) is able, every time an instruction (GMIx) modifying the value of a guard indication is stored in the second memory, also to store the current value of the overflow bit (WROVFL2) of the second write counter in the auxiliary field (BAx) of the second corresponding individual register, and in that the sixth control unit (MC1) includes second auxiliary comparison means (PLO1) able to compare the current value of the overflow bit (RDOVFL2) of the second read counter with the content of the auxiliary field (BAx).

21. Processor according to claim 20, characterized in that the second auxiliary comparison means include an EXCLUSIVE NOR logic gate (PLO1).

22. Processor according to one of claims 19 to 21, characterized in that each stage (ETGi) of the third memory (SNDGQ) includes a field (CHUi) which is usable for the storage of a transmission instruction (SNDGx), and a first supplementary field (Sndgtag), in that the second control means (MCTL) include

a seventh control unit (MC2) able, every time a transmission instruction is stored in the usable field (CHU1) of the input stage of the third memory, to

transfer the content of the main field (CHx) of the second individual register (GTx) associated with the corresponding guard indication (Gx) into the first supplementary field (Sndgtag) of the input stage of the third memory, and

an eighth control unit (MC3) able to derive the criterion for extraction of this transmission instruction having reached the header stage (ETG4) of the third memory, and including second comparison means (MC30) able to compare the content of said first supplementary field (Sndgtag) of the header stage with the current value (RDVL2) of the second read counter.

23. Processor according to claim 22 taken in combination with claim 20 or 21, characterized in that each stage of the third memory further includes a second supplementary field (Auxgmi), in that the seventh control unit (MC2) is able, every time a transmission instruction is stored in the usable field (CHU1) of the input stage of the third memory, to transfer the content of the auxiliary field (BAx) of the second individual register (GTx) associated with the corresponding guard indication (Gx) into the second supplementary field (Auxgmi) of the input stage of the third memory, and in that the eighth control unit (MC3) includes second supplementary comparison means (PLO3) able to compare the content of the second supplementary field (Auxgmi) with the current value of the overflow bit (RDOVFL2) of the second read counter.

24. Processor according to claim 23, characterized in that the second supplementary comparison means include an EXCLUSIVE NOR logic gate (PLO3).